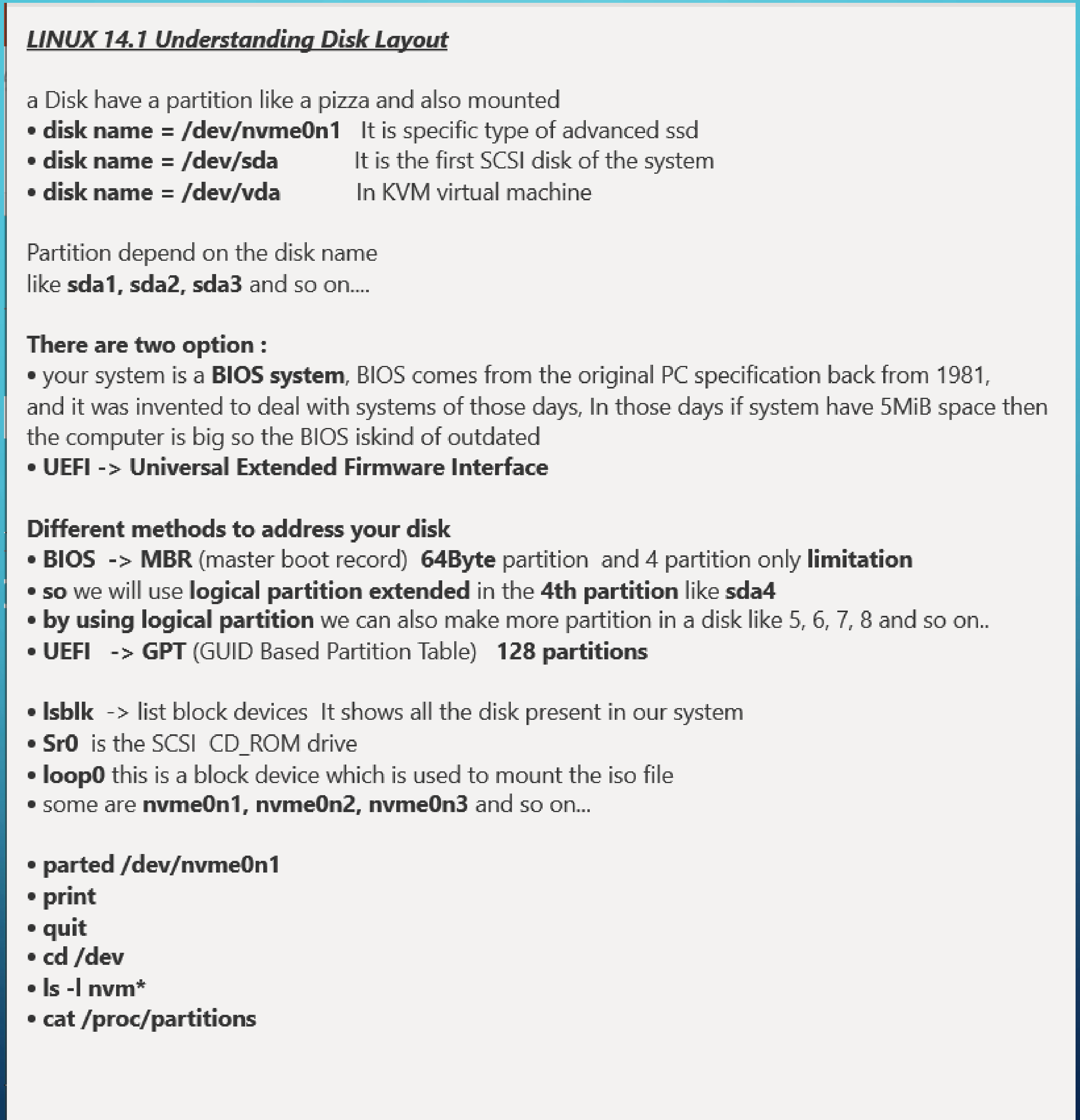
CHAPTER 14

BY :

ETISHA JAIN

VISHU GOYAL



LINUX 14.1 Understanding Disk Layout

a Disk have a partition like a pizza and also mounted

- **disk name = /dev/nvme0n1**   It is specific type of advanced ssd
- **disk name = /dev/sda**   It is the first SCSI disk of the system
- **disk name = /dev/vda**   In KVM virtual machine

Partition depend on the disk name
like **sda1, sda2, sda3** and so on....

There are two option :

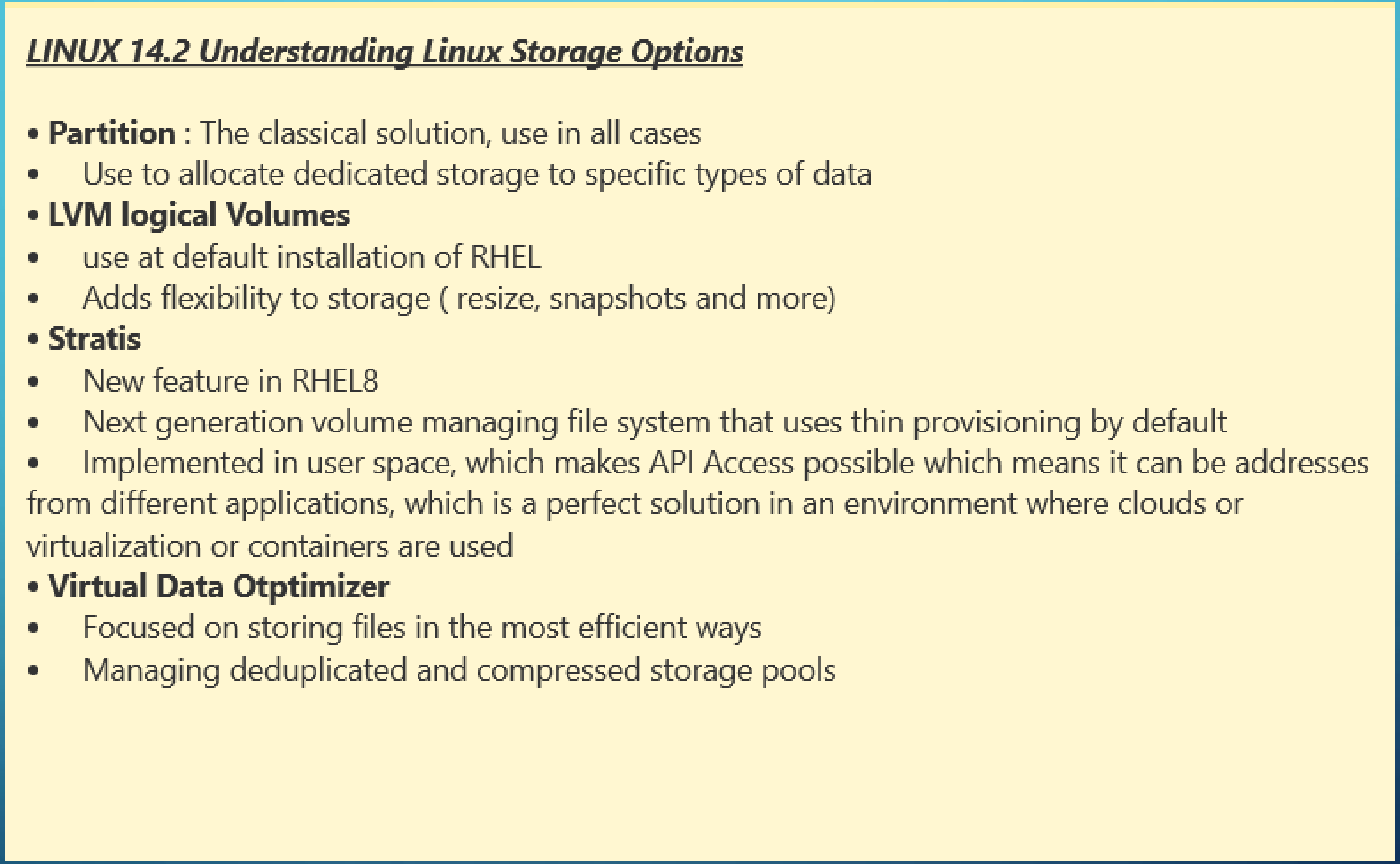
- your system is a **BIOS system**, BIOS comes from the original PC specification back from 1981, and it was invented to deal with systems of those days, In those days if system have 5MiB space then the computer is big so the BIOS iskind of outdated
- **UEFI -> Universal Extended Firmware Interface**

Different methods to address your disk

- **BIOS -> MBR** (master boot record) **64Byte** partition and 4 partition only **limitation**
- **so** we will use **logical partition extended** in the **4th partition** like **sda4**
- **by using logical partition** we can also make more partition in a disk like 5, 6, 7, 8 and so on..
- **UEFI -> GPT** (GUID Based Partition Table) **128 partitions**

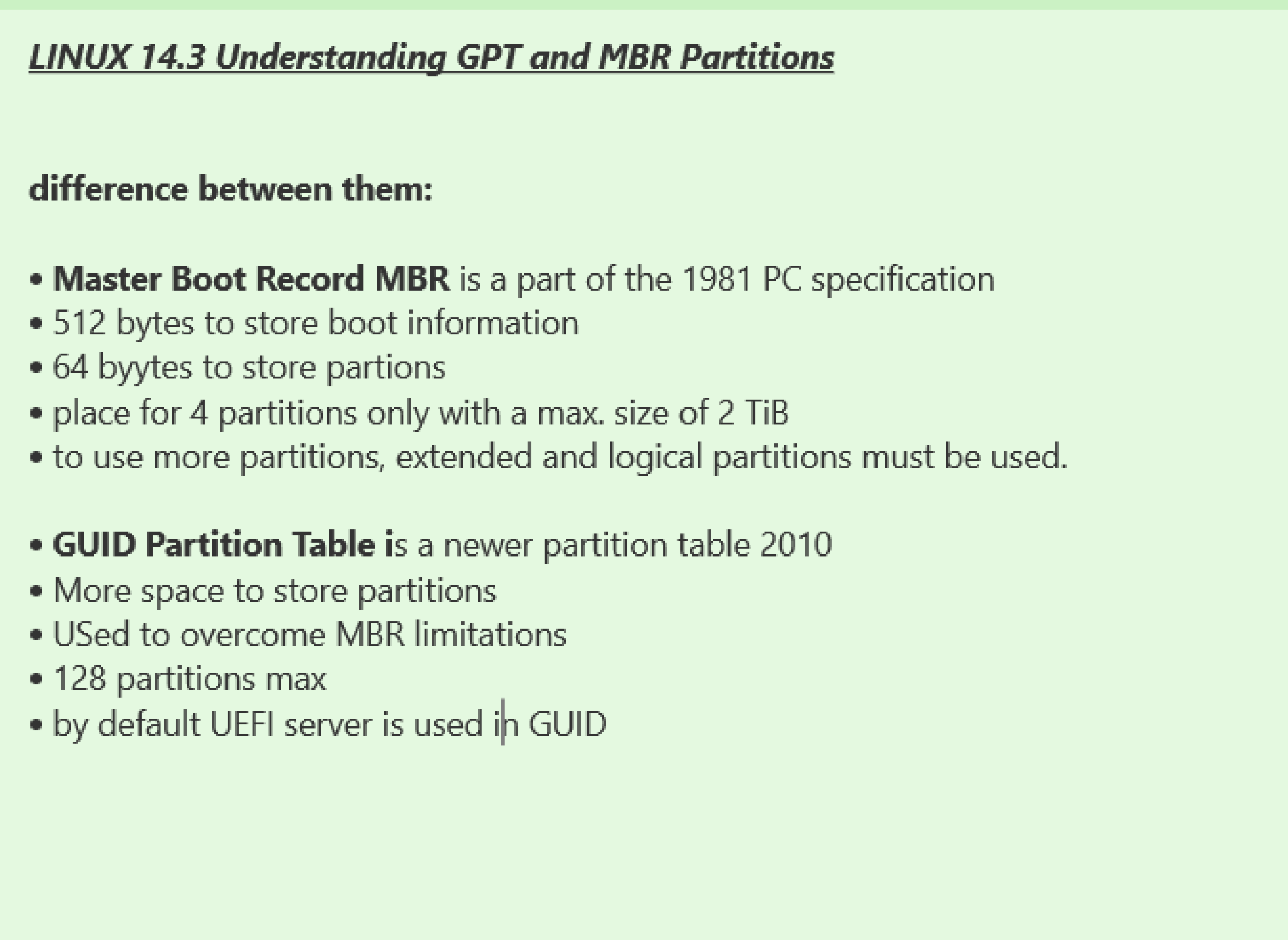
- **lsblk** -> list block devices It shows all the disk present in our system
- **Sr0** is the SCSI CD_ROM drive
- **loop0** this is a block device which is used to mount the iso file
- some are **nvme0n1, nvme0n2, nvme0n3** and so on...

- **parted /dev/nvme0n1**
- **print**
- **quit**
- **cd /dev**
- **ls -l nvm***
- **cat /proc/partitions**



LINUX 14.2 Understanding Linux Storage Options

- **Partition** : The classical solution, use in all cases
- Use to allocate dedicated storage to specific types of data
- **LVM logical Volumes**
- use at default installation of RHEL
- Adds flexibility to storage (resize, snapshots and more)
- **Stratis**
- New feature in RHEL8
- Next generation volume managing file system that uses thin provisioning by default
- Implemented in user space, which makes API Access possible which means it can be addresses from different applications, which is a perfect solution in an environment where clouds or virtualization or containers are used
- **Virtual Data Otptimizer**
- Focused on storing files in the most efficient ways
- Managing deduplicated and compressed storage pools



LINUX 14.3 Understanding GPT and MBR Partitions

difference between them:

- **Master Boot Record MBR** is a part of the 1981 PC specification
- 512 bytes to store boot information
- 64 byytes to store partions
- place for 4 partitions only with a max. size of 2 TiB
- to use more partitions, extended and logical partitions must be used.

- **GUID Partition Table i**s a newer partition table 2010
- More space to store partitions
- USed to overcome MBR limitations
- 128 partitions max
- by default UEFI server is used in GUID

LINUX 14.4 Creating Partitions with parted

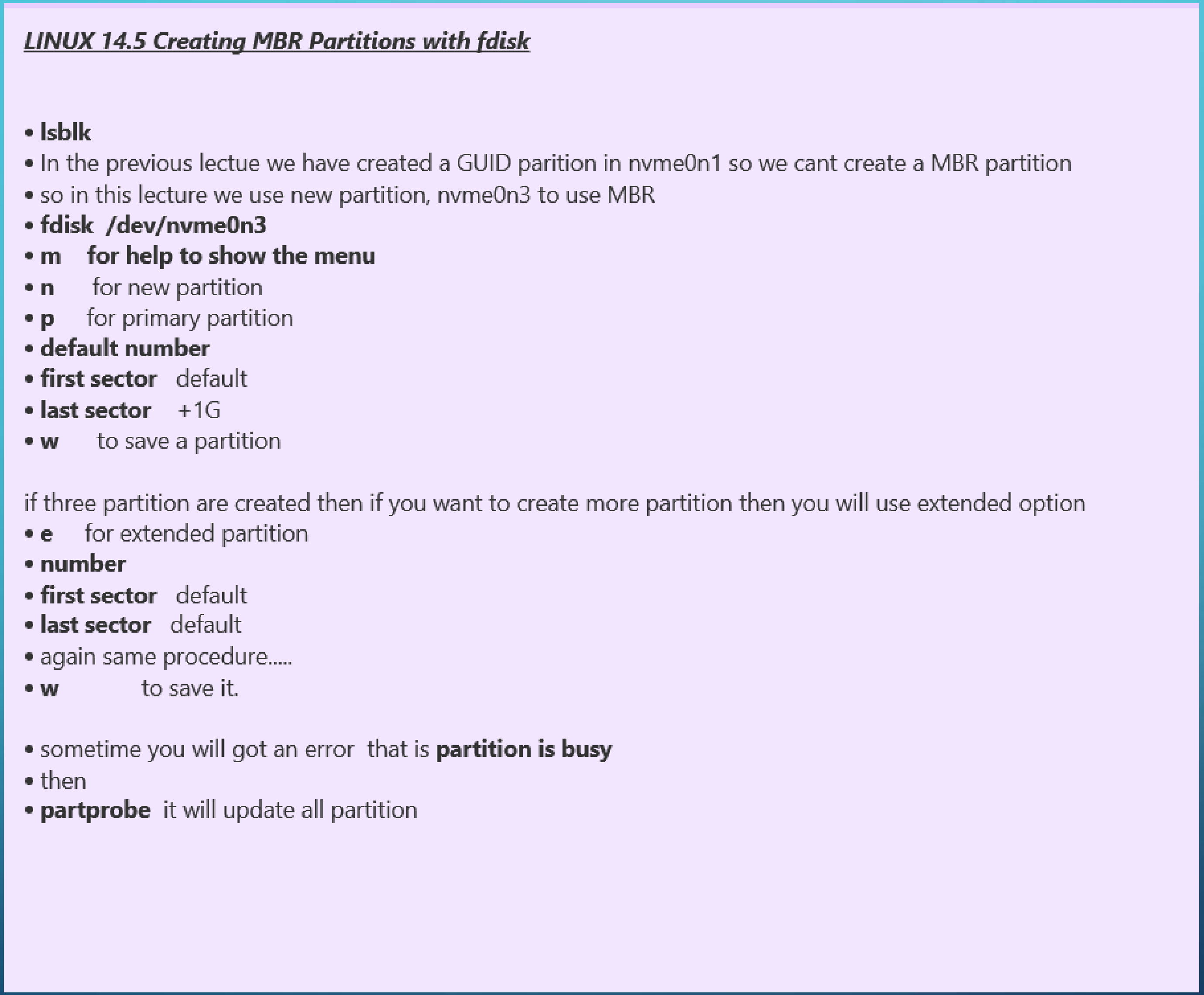
- While **creating a partition**, you do not automatically create a file system
- The **parted file system** attribute only write some unimportant file system metadata
- In RHEL 8, **parted** is defautl utility
- Alternatively, use **fdisk** to work with **MBR** and **gdisk** to use **GUID** partitions

Steps to create a partitions

- **parted /dev/sdb**
- **print** will show if there is a current partition table
- **mklabel msdos | gpt** mklabel is used to set a partitions type and use msdos to set MBR or gpt to create a GUID partition table
- **mkpart part-type name fd-type start end**
- **part-type :** applies to MBR only and sets primary, logical, or extended partition
- **name :** arbitrary name, required for GPT
- **fs-type :** does not modify the filesystem, byt sets someirrelevant file system dependent metadata.
- **start end :** specify start and endm counting from the begining of the disk.
- **for Instance :** mkpart primary 1024MiB 2048MiB.
- Alternatively, use **mkpart** in interactive mode
- **print** to verify creation of the new partition
- **quit** to exit the parted shell
- **udevadm settle** to ensure that the new partition device is created
- **cat /proc/partitions** to verify the creation of the partition

Practical

- lsblk
- parted /dev/nvme0n2
- print
- mklabel gpt
- print
- mkpart one 1MiB 1024 MiB
- print
- quit
- udevadm settle (to update all device)
- cat /proc/partitions
- lsblk



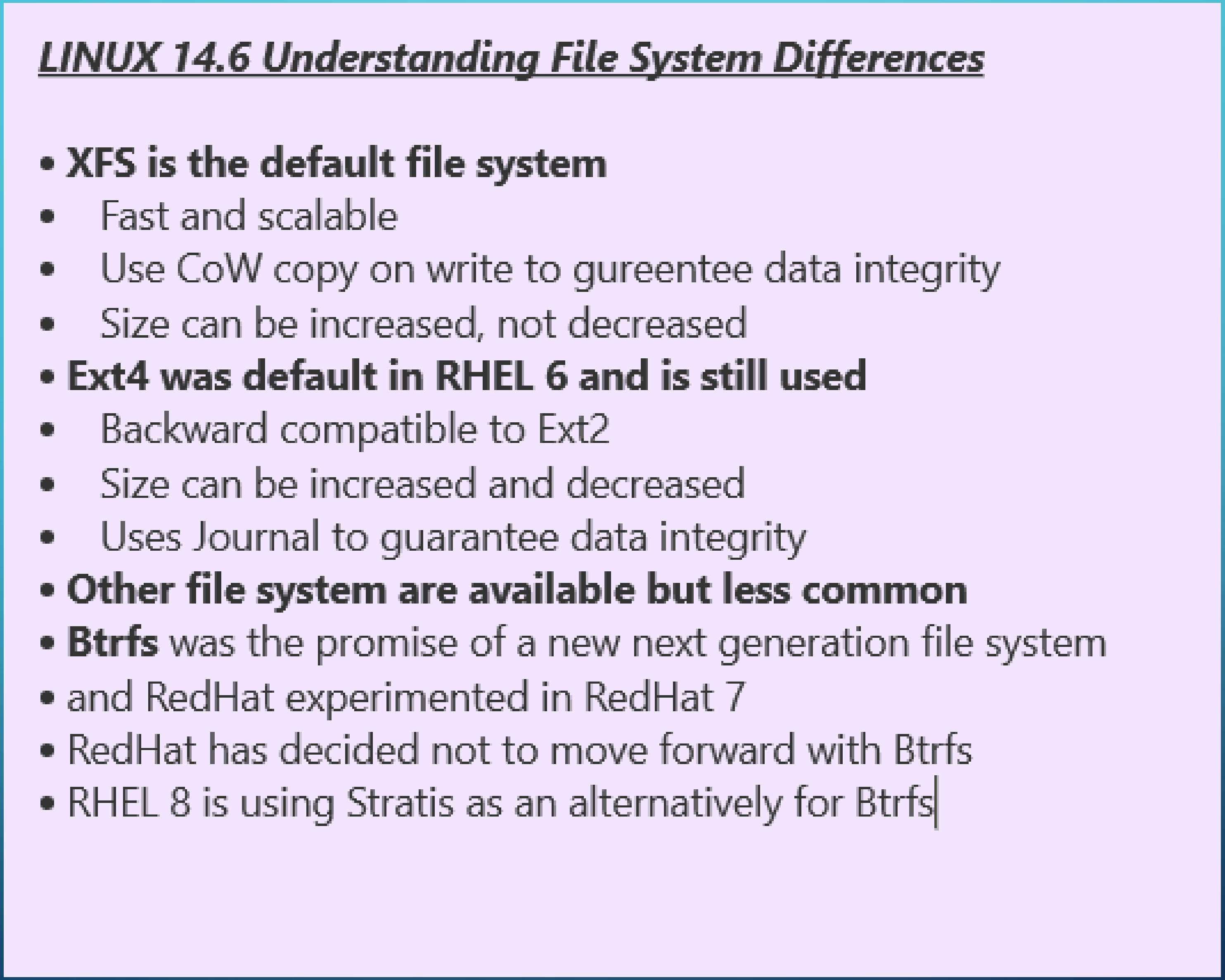
LINUX 14.5 Creating MBR Partitions with fdisk

- **lsblk**
- In the previous lectue we have created a GUID parition in nvme0n1 so we cant create a MBR partition
- so in this lecture we use new partition, nvme0n3 to use MBR
- **fdisk /dev/nvme0n3**
- **m for help to show the menu**
- **n** for new partition
- **p** for primary partition
- **default number**
- **first sector** default
- **last sector** +1G
- **w** to save a partition

if three partition are created then if you want to create more partition then you will use extended option

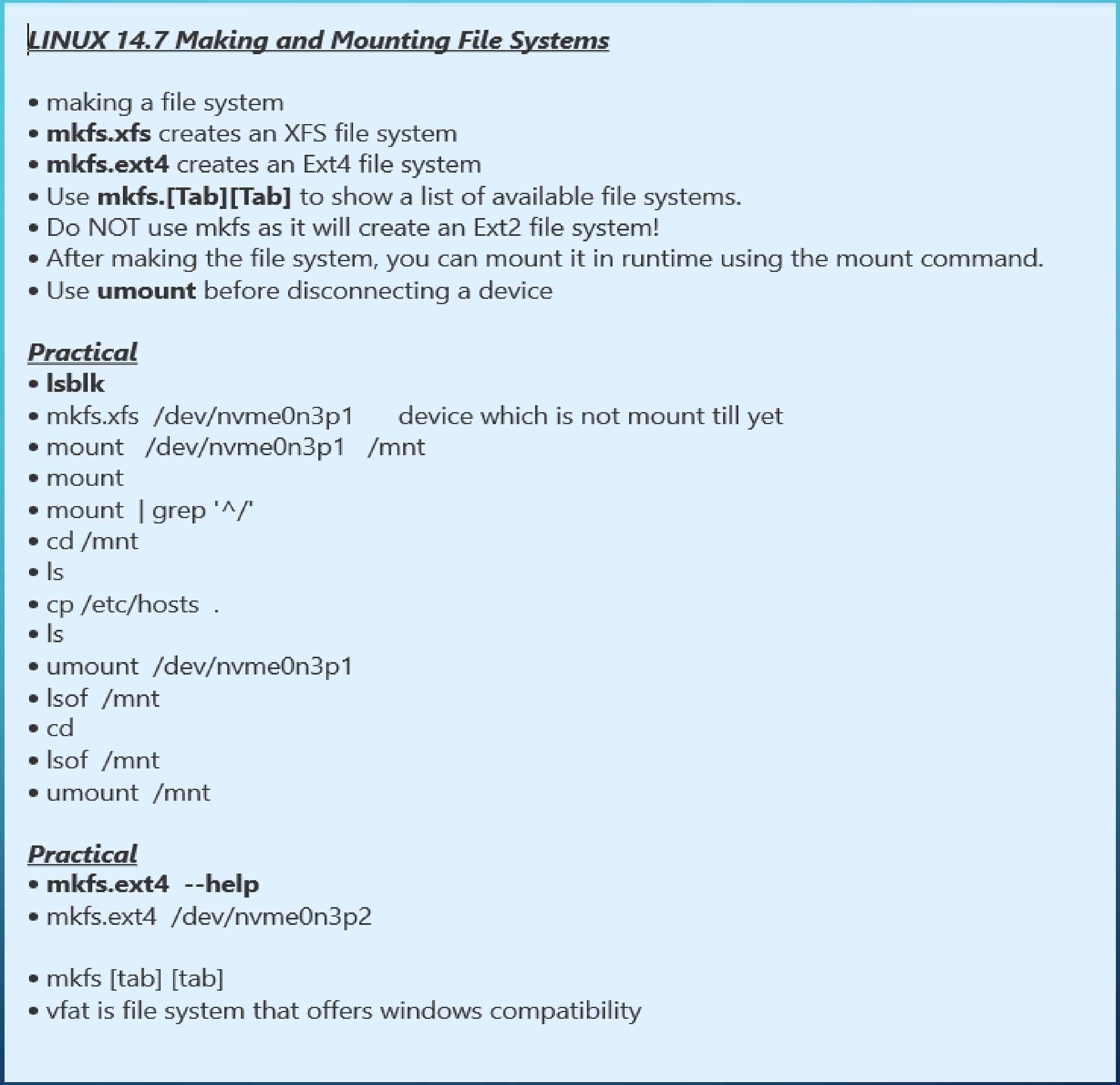
- **e** for extended partition
- **number**
- **first sector** default
- **last sector** default
- again same procedure.....
- **w** to save it.

- sometime you will got an error that is **partition is busy**
- then
- **partprobe** it will update all partition



LINUX 14.6 Understanding File System Differences

- **XFS is the default file system**
- Fast and scalable
- Use CoW copy on write to gureentee data integrity
- Size can be increased, not decreased
- **Ext4 was default in RHEL 6 and is still used**
- Backward compatible to Ext2
- Size can be increased and decreased
- Uses Journal to guarantee data integrity
- **Other file system are available but less common**
- **Btrfs** was the promise of a new next generation file system
- and RedHat experimented in RedHat 7
- RedHat has decided not to move forward with Btrfs
- RHEL 8 is using Stratis as an alternatively for Btrfs



LINUX 14.7 Making and Mounting File Systems

- making a file system
- **mkfs.xfs** creates an XFS file system
- **mkfs.ext4** creates an Ext4 file system
- Use **mkfs.[Tab][Tab]** to show a list of available file systems.
- Do NOT use mkfs as it will create an Ext2 file system!
- After making the file system, you can mount it in runtime using the mount command.
- Use **umount** before disconnecting a device

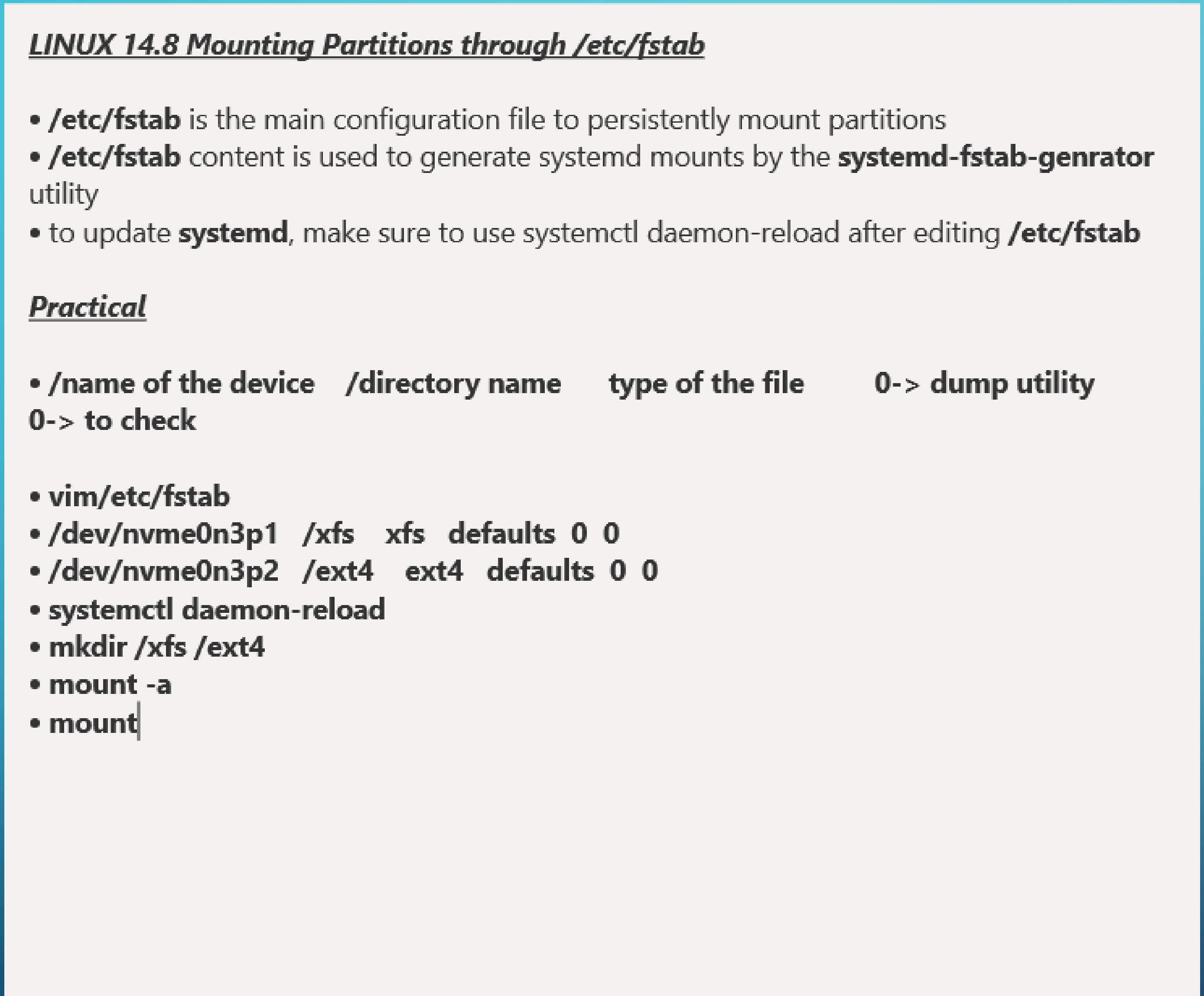
Practical

- **lsblk**
- mkfs.xfs /dev/nvme0n3p1   device which is not mount till yet
- mount   /dev/nvme0n3p1   /mnt
- mount
- mount | grep '^/'
- cd /mnt
- ls
- cp /etc/hosts .
- ls
- umount /dev/nvme0n3p1
- lsof /mnt
- cd
- lsof /mnt
- umount /mnt

Practical

- **mkfs.ext4 --help**
- mkfs.ext4 /dev/nvme0n3p2

- mkfs [tab] [tab]
- vfat is file system that offers windows compatibility



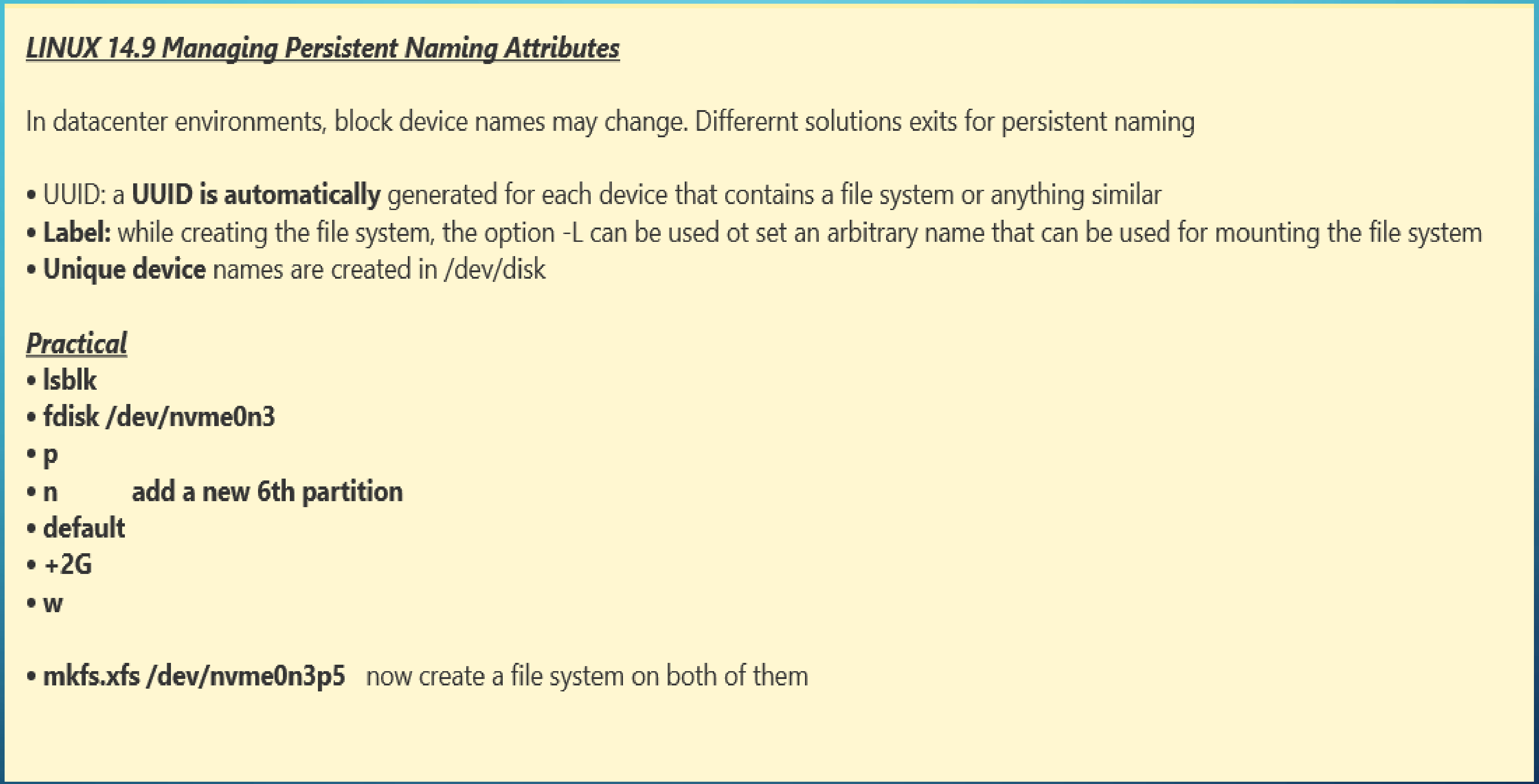
LINUX 14.8 Mounting Partitions through /etc/fstab

- **/etc/fstab** is the main configuration file to persistently mount partitions
- **/etc/fstab** content is used to generate systemd mounts by the **systemd-fstab-genrator** utility
- to update **systemd**, make sure to use systemctl daemon-reload after editing **/etc/fstab**

Practical

- **/name of the device   /directory name   type of the file   0-> dump utility 0-> to check**

- **vim/etc/fstab**
- **/dev/nvme0n3p1   /xfs   xfs   defaults 0 0**
- **/dev/nvme0n3p2   /ext4   ext4   defaults 0 0**
- **systemctl daemon-reload**
- **mkdir /xfs /ext4**
- **mount -a**
- **mount**



LINUX 14.9 Managing Persistent Naming Attributes

In datacenter environments, block device names may change. Differernt solutions exits for persistent naming

- UUID: a **UUID is automatically** generated for each device that contains a file system or anything similar
- **Label:** while creating the file system, the option -L can be used ot set an arbitrary name that can be used for mounting the file system
- **Unique device** names are created in /dev/disk

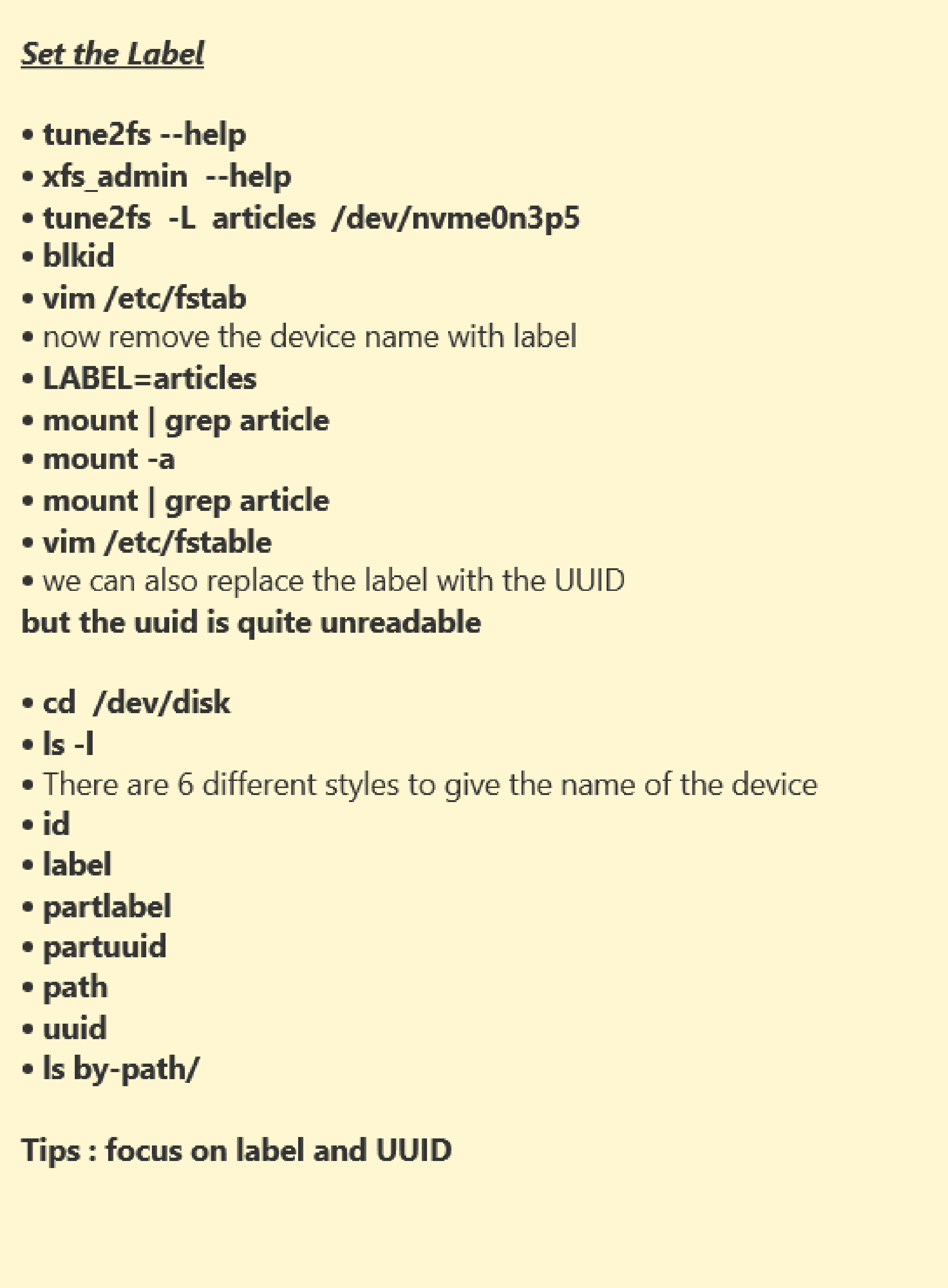
Practical

- **lsblk**
- **fdisk /dev/nvme0n3**
- **p**
- **n   add a new 6th partition**
- **default**
- **+2G**
- **w**

- **mkfs.xfs /dev/nvme0n3p5**   now create a file system on both of them

- **mkfs.ext4 /dev/nvme0n3p6**
- **cd /**
- **mkdir /books /article**
- **vim /etc/fstab**
- **/dev/nvme0n3p5   /books   xfs   defualts   0   0**
- **/dev/nvme0n3p6   /articles   ext4   defualts   0   0**
- **mount -a**
- **mount**

- **fdisk /dev/nvme0n3**
- **d**
- number **5**
- failed to remove device is busy
- **vim /etc/fstab**
- remove that line number for 5th partition or commented
- **reboot**
- press **e** to edit the boot promt
- now remove   **rhgb quiet**
- **press ctrl x**
- **by** default the logical partition starts from 5
- press root pwd
- press **ctrl D**
- **vim /etc/fstab**
- remove or comment that line
- lsblk
- blkid



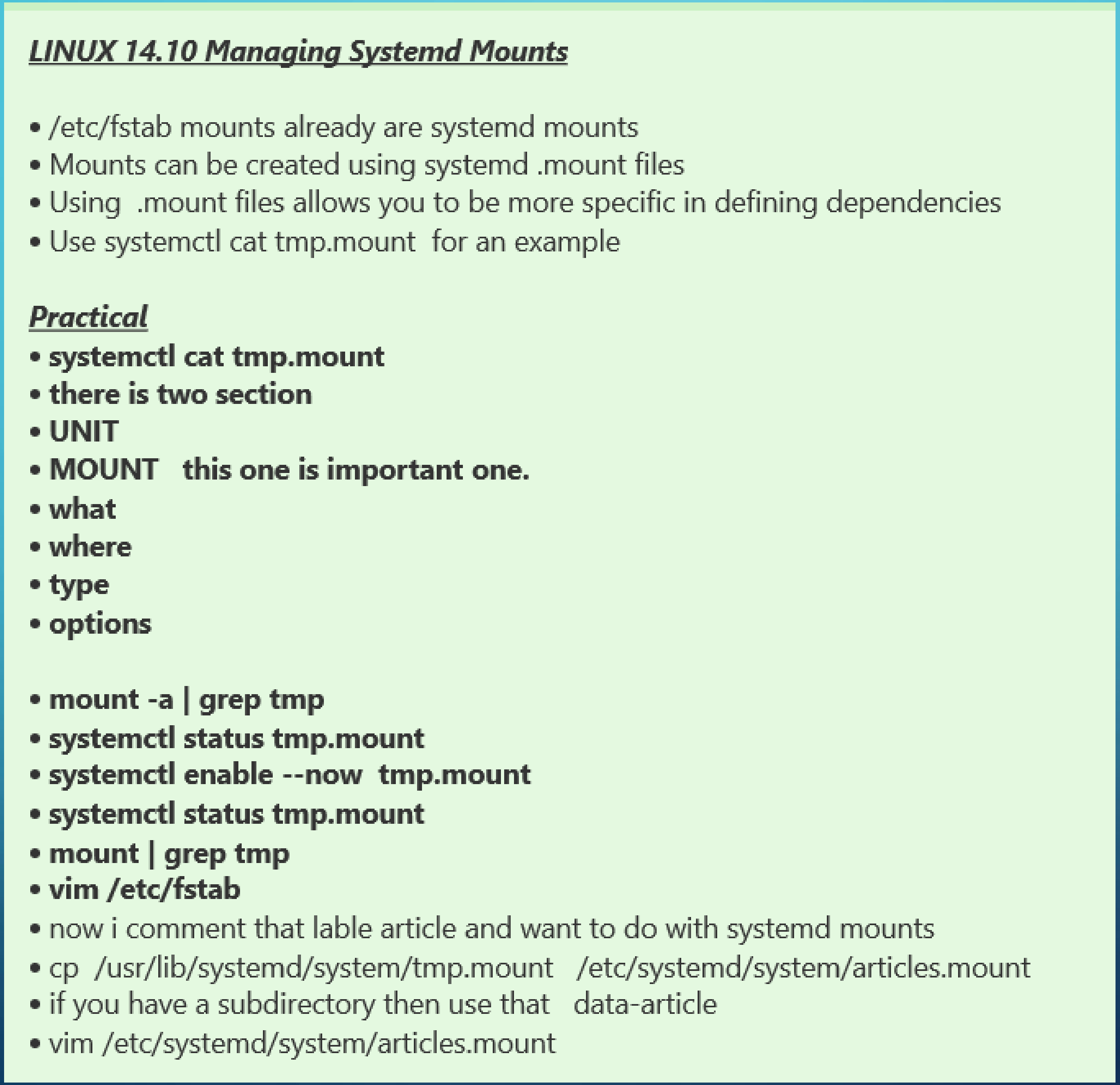
Set the Label

- **tune2fs --help**
- **xfs_admin --help**
- **tune2fs -L articles /dev/nvme0n3p5**
- **blkid**
- **vim /etc/fstab**
- now remove the device name with label
- **LABEL=articles**
- **mount | grep article**
- **mount -a**
- **mount | grep article**
- **vim /etc/fstable**
- we can also replace the label with the UUID

but the uuid is quite unreadable

- **cd /dev/disk**
- **ls -l**
- There are 6 different styles to give the name of the device
- **id**
- **label**
- **partlabel**
- **partuuid**
- **path**
- **uuid**
- **ls by-path/**

Tips : focus on label and UUID



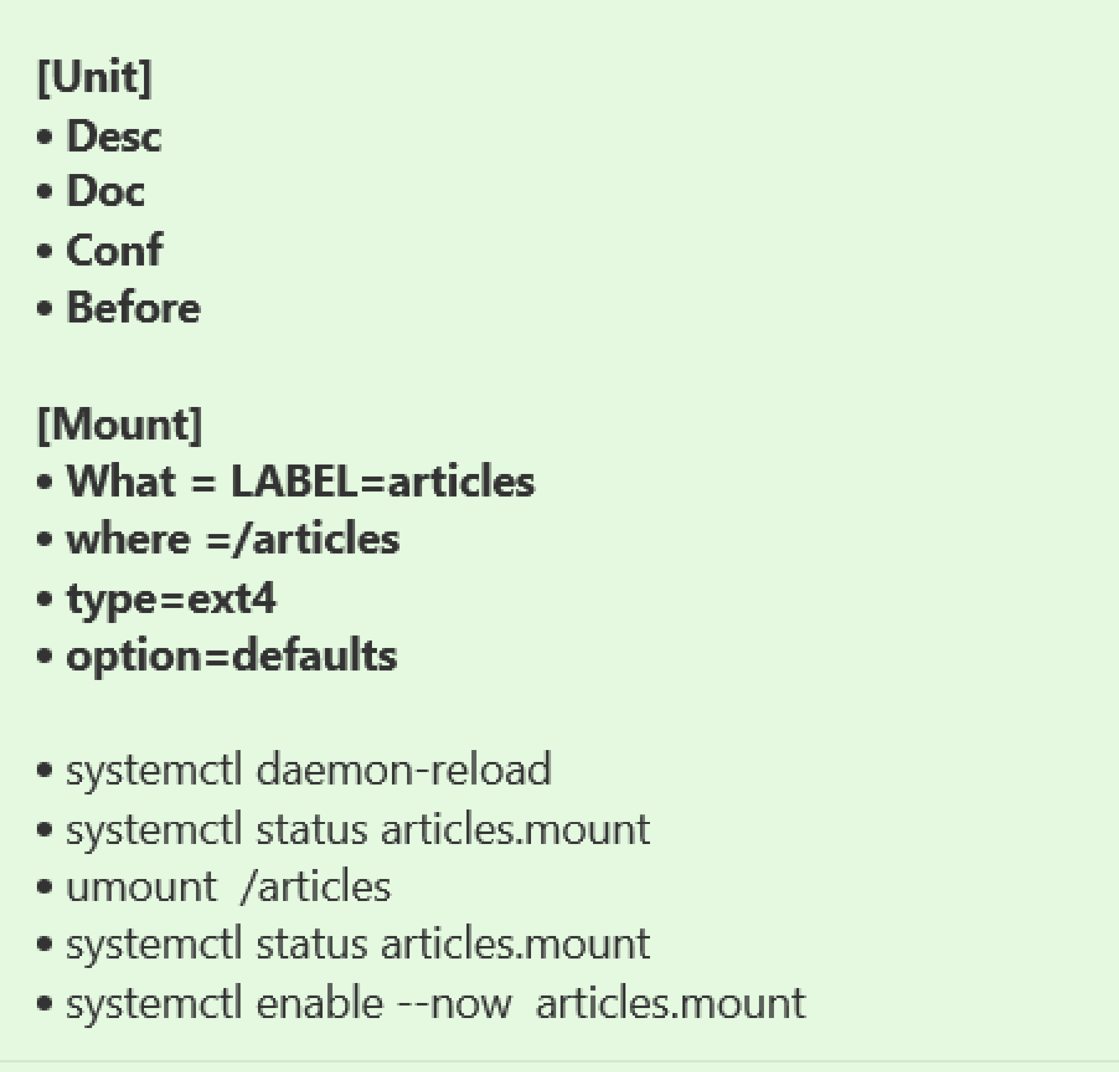
LINUX 14.10 Managing Systemd Mounts

- /etc/fstab mounts already are systemd mounts
- Mounts can be created using systemd .mount files
- Using .mount files allows you to be more specific in defining dependencies
- Use systemctl cat tmp.mount for an example

Practical

- **systemctl cat tmp.mount**
- **there is two section**
- **UNIT**
- **MOUNT this one is important one.**
- **what**
- **where**
- **type**
- **options**

- **mount -a | grep tmp**
- **systemctl status tmp.mount**
- **systemctl enable --now tmp.mount**
- **systemctl status tmp.mount**
- **mount | grep tmp**
- **vim /etc/fstab**
- now i comment that lable article and want to do with systemd mounts
- cp /usr/lib/systemd/system/tmp.mount /etc/systemd/system/articles.mount
- if you have a subdirectory then use that data-article
- vim /etc/systemd/system/articles.mount



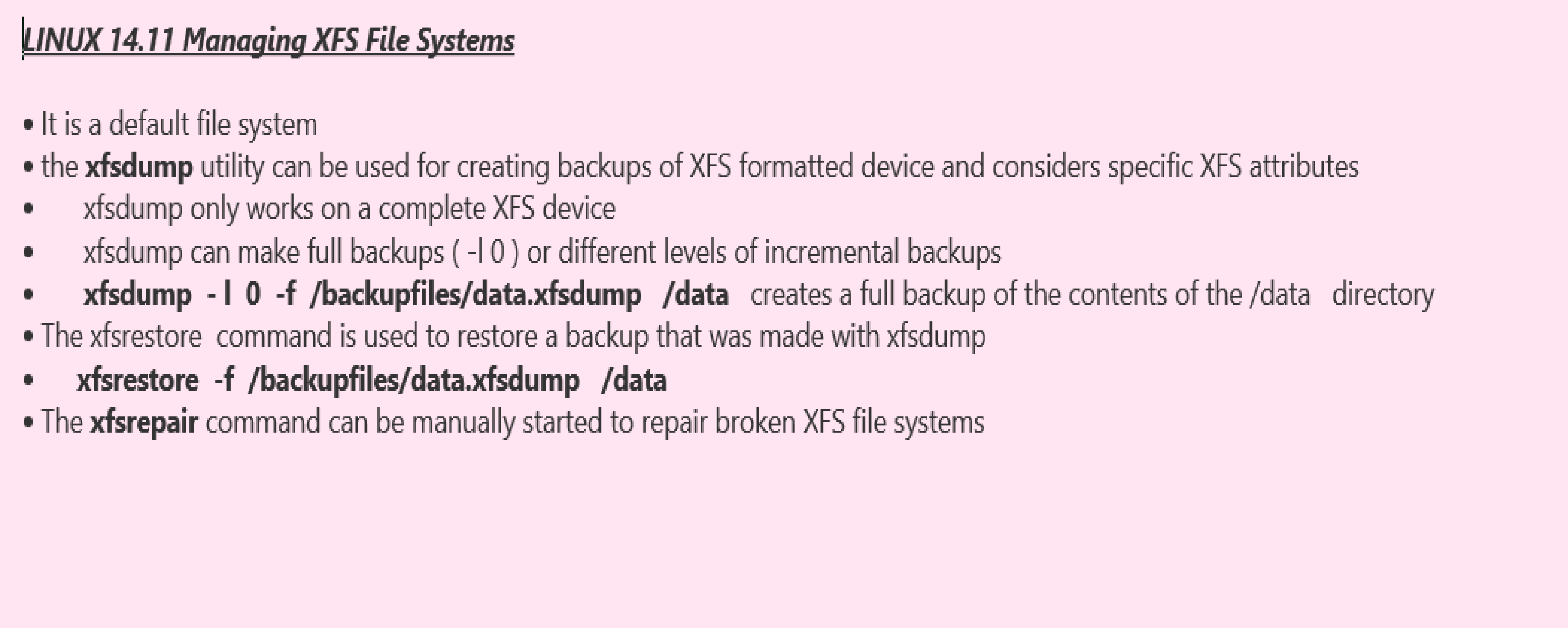
[Unit]

- **Desc**
- **Doc**
- **Conf**
- **Before**

[Mount]

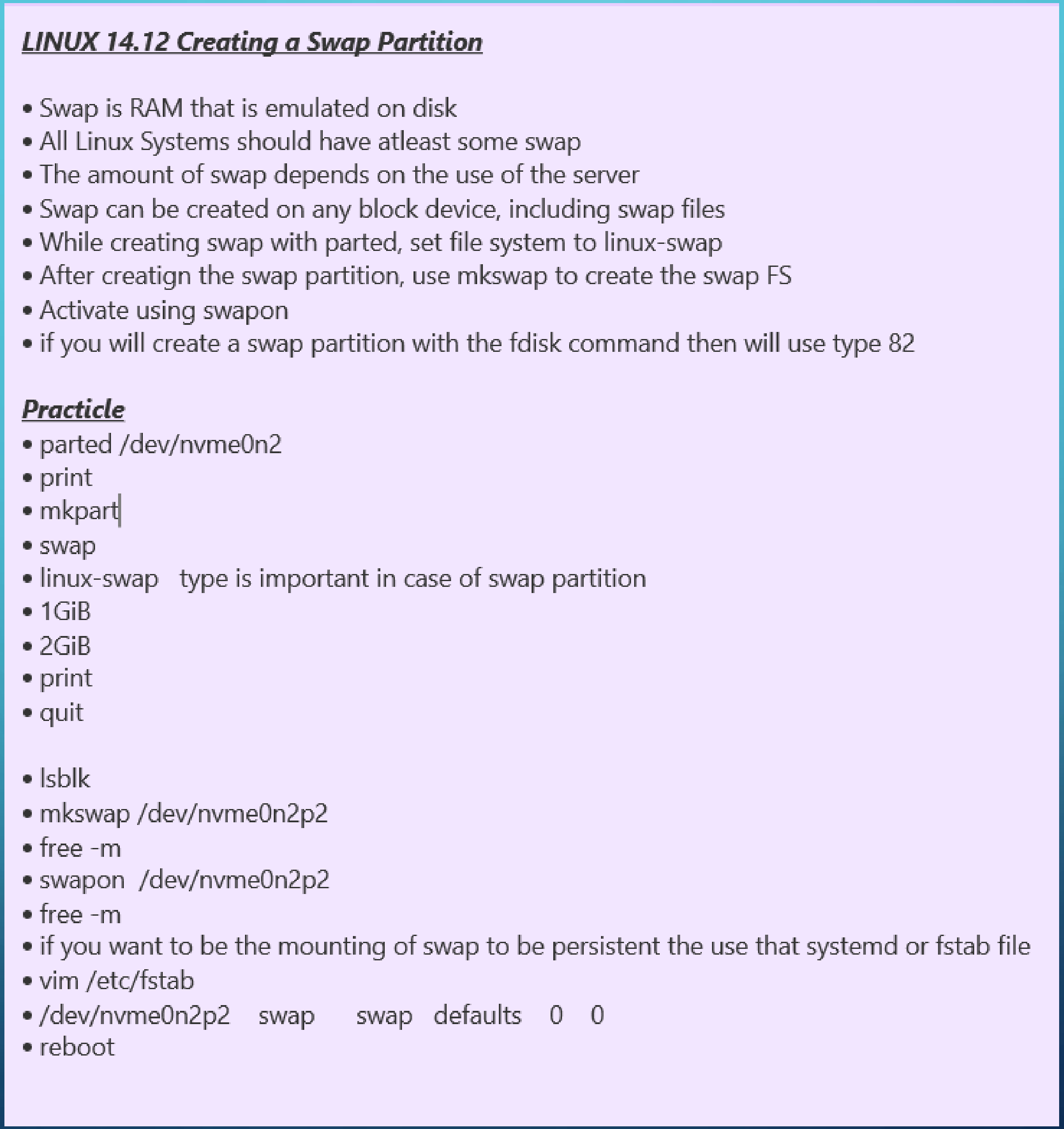
- **What = LABEL=articles**
- **where =/articles**
- **type=ext4**
- **option=defaults**

- systemctl daemon-reload
- systemctl status articles.mount
- umount /articles
- systemctl status articles.mount
- systemctl enable --now articles.mount



LINUX 14.11 Managing XFS File Systems

- It is a default file system
- the **xfsdump** utility can be used for creating backups of XFS formatted device and considers specific XFS attributes
- xfsdump only works on a complete XFS device
- xfsdump can make full backups (-l 0) or different levels of incremental backups
- **xfsdump - l 0 -f /backupfiles/data.xfsdump /data** creates a full backup of the contents of the /data directory
- The xfsrestore command is used to restore a backup that was made with xfsdump
- **xfsrestore -f /backupfiles/data.xfsdump /data**
- The **xfsrepair** command can be manually started to repair broken XFS file systems



LINUX 14.12 Creating a Swap Partition

- Swap is RAM that is emulated on disk
- All Linux Systems should have atleast some swap
- The amount of swap depends on the use of the server
- Swap can be created on any block device, including swap files
- While creating swap with parted, set file system to linux-swap
- After creatign the swap partition, use mkswap to create the swap FS
- Activate using swapon
- if you will create a swap partition with the fdisk command then will use type 82

Practicle

- parted /dev/nvme0n2
- print
- mkpart
- swap
- linux-swap   type is important in case of swap partition
- 1GiB
- 2GiB
- print
- quit

- lsblk
- mkswap /dev/nvme0n2p2
- free -m
- swapon /dev/nvme0n2p2
- free -m
- if you want to be the mounting of swap to be persistent the use that systemd or fstab file
- vim /etc/fstab
- /dev/nvme0n2p2   swap   swap   defaults   0   0
- reboot

THANK YOU